Reservation Strategies for Stochastic Jobs

Guillaume Aupy[1], Ana Gainaru[2], Valentin Honoré[1],
Padma Raghavan[2], Yves Robert[3], Hongyang Sun[2]

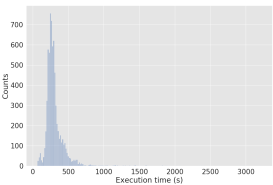
1. Inria & Univ Bordeaux; 2. Vanderbilt University;
3. ENS Lyon & UTK

Datamove & Polaris Seminar, december 2018

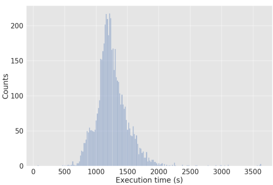
informatiques mathématiques

Motiv: Neuroscience Applications

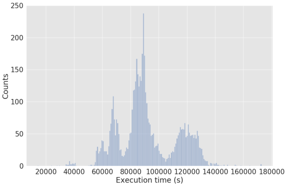
Often the execution time of an application is unknown before it runs.



(a) fMRIQA



(b) VBMQA



(c) dtiQA

Figure: Traces [2013-2016] of neuroscience apps (Vanderbilt's medical imaging database).

These applications are input dependent, but predicting the exact execution time is hard even when knowing the input.

Motiv: Computing in the cloud

Several cost models to compute in the cloud:

- **On-Demand (OD):** "you pay for compute capacity by per hour or per second depending on which instances you run" (Amazon AWS).

  (= Pay what you use)

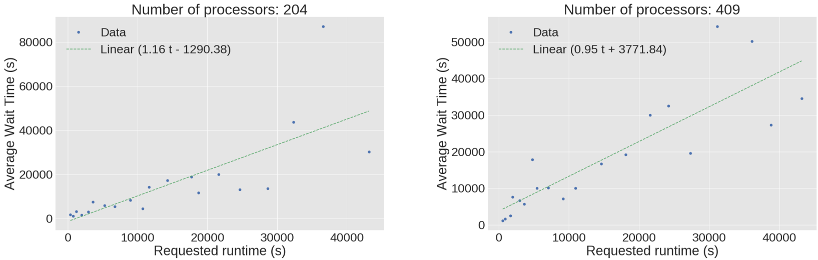
- **Reserved-Instances (RI):** "Reserved Instances provide you with a significant discount (up to 75%) compared to On-Demand instance pricing."

  (= Pay what you ask for)

Payment Option	Upfront	Monthly*	Effective Hourly**	Savings over On-Demand	On-Demand Hourly
No Upfront	$0.00	$8.03	$0.011	57%	
Partial Upfront	$134.00	$3.72	$0.01	60%	$0.0255
All Upfront	$252.00	$0.00	$0.01	62%	

Figure: Data extracted from AWS website, 12/10/2018.

Execution time = Wait time + Runtime:



(a) Jobs that requested 204 procs. (b) Jobs that requested 409 procs.

Figure: Average wait times of jobs run on Intrepid (2009) as a function of requested runtime (data: Parallel Workload Archive).

Observation: For a given number of processors, wait time is an increasing function of requested time.

Reservation Strategies for Stochastic Jobs

- Job execution time follows a Random Variable X.
  - Distribution $\mathcal{D}$
  - Cumulative function (CDF) F ($F(x) = \mathbb{P}(X \leq x)$)*
  - Density function (PDF) f
  - Support is positive ($X \in [\min_{\mathcal{D}}, \max_{\mathcal{D}}]$, s.t. $\min_{\mathcal{D}} \geq 0$ and $\max_{\mathcal{D}} \in \mathbb{R} \cup \{\infty\}$)

- Deterministic jobs (two executions of the same job have the same duration).

*most of the results assume a smooth CDF

Platform cost model

We consider a platform where one can request time:

- Assume a user **reserves** a set of resource for a time T;
- The resources are **needed** for a time t.

$$\textbf{Cost: } \alpha T + \beta \min(T, t) + \gamma \tag{1}$$

α, β, γ = platform parameters

We consider a platform where one can request time:

- Assume a user **reserves** a set of resource for a time T;
- The resources are **needed** for a time t.

$$\textbf{Cost: } \alpha T + \beta \min(T, t) + \gamma \qquad (1)$$

α, β, γ = platform parameters

- Reservation cost: what is paid for the reservation.

Platform cost model

We consider a platform where one can request time:

- Assume a user **reserves** a set of resource for a time T;
- The resources are **needed** for a time t.

$$\textbf{Cost: } \alpha T + \beta \min(T, t) + \gamma \qquad (1)$$

α, β, γ = platform parameters

- Reservation cost: what is paid for the reservation.
- Utilization cost: what is paid for the usage.

Platform cost model

We consider a platform where one can request time:

- Assume a user **reserves** a set of resource for a time T;
- The resources are **needed** for a time t.

$$\textbf{Cost:}\ \alpha T + \beta \min(T, t) + \gamma \quad (1)$$

α, β, γ = platform parameters (e.g., OD: $\alpha = \gamma = 0$, RI: $\beta = \gamma = 0$)

- Reservation cost: what is paid for the reservation.
- Utilization cost: what is paid for the usage.

Reservation-based Approach

Given a job J of duration $\mathbf{t}$ (unknown). The user makes a reservation of time t_1. Two cases:

- $\mathbf{t} \leq t_1$ The reservation is enough and the job succeeds.
- $\mathbf{t} > t_1$ The reservation is not enough. The job fails. The user needs to ask for another reservation $t_2 > t_1$.

Given a job J of duration $\mathbf{t}$ (unknown). The user makes a reservation of time t_1. Two cases:

- $\mathbf{t} \leq t_1$ The reservation is enough and the job succeeds.
- $\mathbf{t} > t_1$ The reservation is not enough. The job fails. The user needs to ask for another reservation $t_2 > t_1$.

Let $S = (t_1, t_2, \ldots, t_{k-1}, t_k)$ be a sequence of reservations, s.t. $t_1 < ... < t_{k-1} < \mathbf{t} \leq t_k$. The cost to the user is:

$$C(k, t) = \sum_{i=1}^{k-1} (\alpha t_i + \beta t_i + \gamma) + \alpha t_k + \beta t + \gamma$$

Optimization Problem

Given a sequence of increasing reservation $S = (t_1, t_2, \dots, t_i, t_{i+1}, \dots)$.
Given a distribution of execution time X (PDF f). The expected cost of S on X is:

$$\mathbb{E}(S) = \sum_{k=1}^{\infty} \int_{t_{k-1}}^{t_k} C(k,t) f(t) dt \qquad (2)$$

Definition (Stochastic)

Given a stochastic job of execution time X. Given α, β, γ defining a cost function (Eq. (1)), find a reservation sequence S with minimal expected cost $\mathbb{E}(S)$ (Eq. (2)).

Take-aways

- (Almost) Caracterization of optimal sequence for **any** distribution!
- RI strategy is in general useful, even for stochastic job!
- Comparison to "natural" greedy strategies show the importance of the optimal sequence.

Take-aways

- (Almost) Caracterization of optimal sequence for **any** distribution!
- RI strategy is in general useful, even for stochastic job!
- Comparison to “natural” greedy strategies show the importance of the optimal sequence.

Additionally, opens up a whole venue of new fun scheduling problems ☺

Cost function

As a preliminary lemma (warm up):

$$\mathbb{E}(S) = \sum_{k=1}^{\infty} \int_{t_{k-1}}^{t_k} C(k,t) f(t) dt$$
$$= \boxed{\beta \cdot \mathbb{E}[X] + \sum_{i=0}^{\infty} (\alpha t_{i+1} + \beta t_i + \gamma) \, \mathbb{P}(X \geq t_i)}$$

Reminder: $t_0 = 0$; $C(k,t) = \sum_{i=1}^{k-1} (\alpha t_i + \beta t_i + \gamma) + \alpha t_k + \beta t + \gamma$

Existence

Q1: Existence of a solution $S = (t_1, t_2, \dots, t_i, t_{i+1}, \dots)$ with finite expected cost?

Yes[†]! Constructive proof: $(t_i)_i = (\min_{\mathcal{D}} + i)_i$ has expectation $\beta \cdot \mathbb{E}(X) + \alpha T_1 + \gamma$, where

$$T_1 = \mathbb{E}[X] + 1 + \frac{\alpha+\beta}{2\alpha}(\mathbb{E}[X^2] - {\min_{\mathcal{D}}}^2) + \frac{\alpha+\beta+\gamma}{\alpha}(\mathbb{E}[X] - \min_{\mathcal{D}})$$

[†] Conditions: $\mathrm{Var}[X] < \infty$

Existence

Q1: Existence of a solution $S = (t_1, t_2, \ldots, t_i, t_{i+1}, \ldots)$ with finite expected cost?

Yes[†]! Constructive proof: $(t_i)_i = (\min_{\mathcal{D}} + i)_i$ has expectation $\beta \cdot \mathbb{E}(X) + \alpha T_1 + \gamma$, where

$$T_1 = \mathbb{E}[X] + 1 + \frac{\alpha+\beta}{2\alpha}(\mathbb{E}[X^2] - \min_{\mathcal{D}}{}^2) + \frac{\alpha+\beta+\gamma}{\alpha}(\mathbb{E}[X] - \min_{\mathcal{D}})$$

Denote $S^o = (t_1^o, t_2^o, \cdots, t_i^o, \cdots)$ an optimal sequence for STOCHASTIC. $\boxed{\textbf{Coro: } t_1^o \leq T_1}$

[†]Conditions: $\mathrm{Var}[X] < \infty$

Theorem (Optimal algorithm)

For X smooth, Stochastic *reduces to finding t_1^o that minimizes:*

$$\sum_{i=0}^{\infty} (\alpha t_{i+1} + \beta t_i + \gamma)\,\mathbb{P}(X \geq t_i) \qquad \textbf{s.t. for } i \geq 1 \quad F(t_i^o) = 1 \textbf{ or } t_{i+1}^o = g_{X,\alpha,\beta,\gamma}(t_i^o, t_{i-1}^o)$$

Theorem (Optimal algorithm)

For X smooth, Stochastic *reduces to finding t_1^o that minimizes:*

$$\sum_{i=0}^{\infty} (\alpha t_{i+1} + \beta t_i + \gamma)\,\mathbb{P}(X \geq t_i) \qquad \textbf{s.t. for } i \geq 1 \quad F(t_i^o) = 1 \textbf{ or } t_{i+1}^o = g_{X,\alpha,\beta,\gamma}(t_i^o, t_{i-1}^o)$$

- We study: $S(t) = (t_1^o, \cdots, t_{i-1}^o, t, t_{i+1}^o, \cdots)$

Theorem (Optimal algorithm)

For X smooth, Stochastic *reduces to finding t_1^o that minimizes:*

$$\sum_{i=0}^{\infty} (\alpha t_{i+1} + \beta t_i + \gamma)\, \mathbb{P}(X \geq t_i) \qquad \textbf{s.t. for } i \geq 1 \quad F(t_i^o) = 1 \textbf{ or } t_{i+1}^o = g_{X,\alpha,\beta,\gamma}(t_i^o, t_{i-1}^o)$$

- We study: $S(t) = (t_1^o, \cdots, t_{i-1}^o, t, t_{i+1}^o, \cdots)$
- $\mathbb{E}[S(t)]$ on $[t_{i-1}^o, t_{i+1}^o]$ is minimized:
  1. For $t = t_i^o$
  2. For $(\mathbb{E}[S(t)])' = 0$ (if $F(t_i^o) \neq 1$)

Theorem (Optimal algorithm)

For X smooth, Stochastic *reduces to finding t_1^o that minimizes:*

$$\sum_{i=0}^{\infty} (\alpha t_{i+1} + \beta t_i + \gamma)\, \mathbb{P}(X \geq t_i) \qquad \textbf{s.t. for } i \geq 1 \quad F(t_i^o) = 1 \textbf{ or } t_{i+1}^o = g_{X,\alpha,\beta,\gamma}(t_i^o, t_{i-1}^o)$$

- We study: $S(t) = (t_1^o, \cdots, t_{i-1}^o, t, t_{i+1}^o, \cdots)$
- $\mathbb{E}[S(t)]$ on $[t_{i-1}^o, t_{i+1}^o]$ is minimized:
  1. For $t = t_i^o$
  2. For $(\mathbb{E}[S(t)])' = 0$ (if $F(t_i^o) \neq 1$)
- We get:

$$\alpha t_{i+1}^o + \beta t_i^o + \gamma = \alpha \frac{1 - F(t_{i-1}^o)}{f(t_i^o)} + \beta \frac{1 - F(t_i^o)}{f(t_i^o)}$$

- which enables to define: $g_{X,\alpha,\beta,\gamma}(t_i^o, t_{i-1}^o)$.

Optimal sequence

Theorem (Optimal algorithm)

For X smooth, Stochastic *reduces to finding t_1^o that minimizes:*

$$\sum_{i=0}^{\infty} (\alpha t_{i+1} + \beta t_i + \gamma)\, \mathbb{P}(X \geq t_i) \qquad \textbf{s.t. for } i \geq 1 \quad F(t_i^o) = 1 \textbf{ or } t_{i+1}^o = g_{X,\alpha,\beta,\gamma}(t_i^o, t_{i-1}^o)$$

- We study: $S(t) = (t_1^o, \cdots, t_{i-1}^o, t, t_{i+1}^o, \cdots)$
- $\mathbb{E}[S(t)]$ on $[t_{i-1}^o, t_{i+1}^o]$ is minimized:
  1. For $t = t_i^o$
  2. For $(\mathbb{E}[S(t)])' = 0$ (if $F(t_i^o) \neq 1$)
- We get:

$$\alpha t_{i+1}^o + \beta t_i^o + \gamma = \alpha \frac{1 - F(t_{i-1}^o)}{f(t_i^o)} + \beta \frac{1 - F(t_i^o)}{f(t_i^o)}$$

- which enables to define: $g_{X,\alpha,\beta,\gamma}(t_i^o, t_{i-1}^o)$.

We are still missing t_1^o! (by convention we set $t_0^o = 0$)

Brute-Force procedure for t_1^o

We know: $t_1^o \in [\min_{\mathcal{D}}; T_1]$ (T_1 = upper bound from earlier).

- Let $S_{\mathrm{bf}}(t_1) = (t_0, t_1, \cdots, t_i, \cdots)$ s.t.:

$$\begin{cases} t_0 = 0, \\ t_i = g_{X,\alpha,\beta,\gamma}(t_{i-1}, t_{i-2}) \quad (i \geq 2). \end{cases}$$

Brute-Force procedure for t_1^o

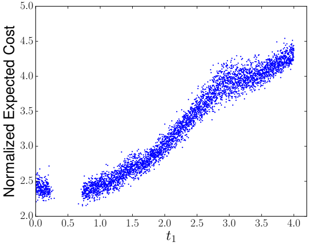
We know: $t_1^o \in [\min_{\mathcal{D}}; T_1]$ (T_1 = upper bound from earlier).

- Let $S_{\text{bf}}(t_1) = (t_0, t_1, \cdots, t_i, \cdots)$ s.t.:

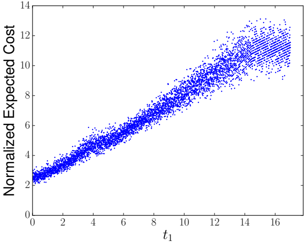
$$\begin{cases} t_0 = 0, \\ t_i = g_{X,\alpha,\beta,\gamma}(t_{i-1}, t_{i-2}) \quad (i \geq 2). \end{cases}$$

Brute-force procedure:

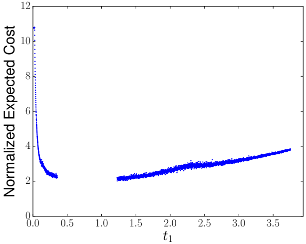
- $t_1 = \min_{\mathcal{D}} + m \cdot \frac{T_1 - \min_{\mathcal{D}}}{M}, \forall m = 1, \cdots, M$ (in practice we chose $M = 5000$):
- Expected cost of each sequence via a Monte-Carlo process
  - Randomly draw N execution time from distribution ($N = 1000$ here)
  - Evaluate average cost over N samples of $S_{\text{bf}}(t_1)$ ($\mathcal{O}(MN)$)
- Keep the best: t_1^{algo}.



(a) Exponential Distribution



(b) Weibull Distribution



(c) Gamma Distribution

Figure: Monte-Carlo simulations, normalized $=$ Cost$/\mathbb{E}(X)$, $\alpha = 1$, $\beta = \gamma = 0$.

In practice, Monte-Carlo simulations

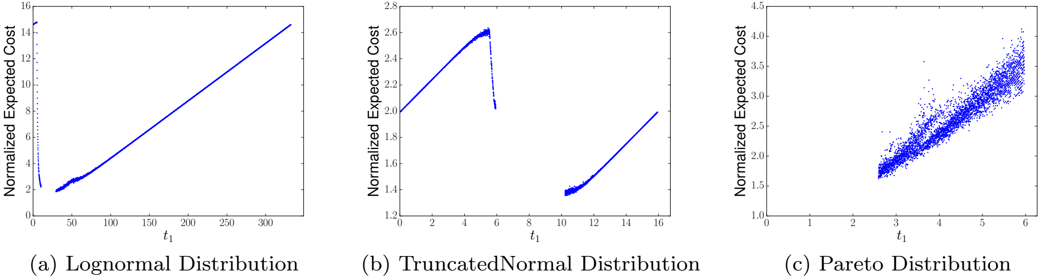
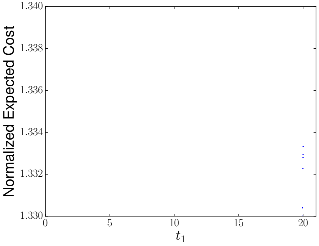
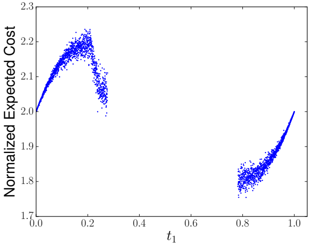


Figure: Monte-Carlo simulations, normalized $=$ Cost$/\mathbb{E}(X)$, $\alpha = 1$, $\beta = \gamma = 0$.

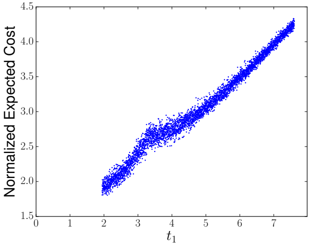
In practice, Monte-Carlo simulations



(a) Uniform Distribution



(b) Beta Distribution



(c) BoundedPareto Distribution

Figure: Monte-Carlo simulations, normalized $= \text{Cost}/\mathbb{E}(X)$, $\alpha = 1$, $\beta = \gamma = 0$.

Dynamic programming algorithm

Theorem

If $X \sim (v_i, f_i)_{i=1\ldots n}$ a discrete distribution, Stochastic *can be solved in polynomial time.*

- Dynamic Programming algorithm:

$$\mathbb{E}_i^* = \min_{i \le j \le n} \left(\alpha v_j + \gamma + \sum_{k=i}^{j} f'_k \cdot \beta v_k + \Big(\sum_{k=j+1}^{n} f'_k \Big) \big(\beta v_j + \mathbb{E}_{j+1}^* \big) \right)$$

$$f'_k = \frac{f_k}{\sum_{j=i}^{n} f_j}, \forall k = i, \ldots, n$$

$$\mathbb{E}_n^* = \alpha v_n + \beta v_n + \gamma$$

- Complexity: $\mathcal{O}(n^2)$

Truncation and discretization

To use the previous Theorem, we can discretize a continuous probability distribution.

- Truncation + Discretization: Given a precision ε;
  - Change support to $[\min_{\mathcal{D}}, Q(1-\varepsilon)^{\ddagger}]$ (for infinite support).
  - Discretize the support: n discrete values: $(v_i, f_i)_{i=1\ldots n}$

$^{\ddagger}Q(x) = \inf\{t | F(t) \geq x\}$

Truncation and discretization

To use the previous Theorem, we can discretize a continuous probability distribution.

- Truncation + Discretization: Given a precision ε;
  - Change support to $[\min_{\mathcal{D}}, Q(1-\varepsilon)^{\ddagger}]$ (for infinite support).
  - Discretize the support: n discrete values: $(v_i, f_i)_{i=1\dots n}$
- Two discretization schemes

Equal-probability:	all discrete chunks have same probability
Equal-time:	all chunks equally spaced in $[\min_{\mathcal{D}}, Q(1-\varepsilon)]$

$^{\ddagger} Q(x) = \inf\{t | F(t) \geq x\}$

Truncation and discretization

To use the previous Theorem, we can discretize a continuous probability distribution.

- Truncation + Discretization: Given a precision ε;
  - Change support to $[\min_{\mathcal{D}}, Q(1-\varepsilon)^{\ddagger}]$ (for infinite support).
  - Discretize the support: n discrete values: $(v_i, f_i)_{i=1\dots n}$
- Two discretization schemes

  Equal-probability: all discrete chunks have same probability

  Equal-time: all chunks equally spaced in $[\min_{\mathcal{D}}, Q(1-\varepsilon)]$
- Evaluation: $\varepsilon = 10^{-7}$, $n = 1000$

$^{\ddagger}Q(x) = \inf\{t | F(t) \geq x\}$

- $\mu = \mathbb{E}(X) = \int_0^\infty t f(t) dt \qquad \sigma^2 = \mathbb{E}(X^2) - \mu^2$
- 4 different heuristics
  - Mean-by-Mean:
  $$t_i = \mathbb{E}(X | X > t_{i-1}) = \frac{\int_{t_{i-1}}^\infty t f(t) dt}{1 - F(t_{i-1})},\ \forall i \geq 2$$
  - Mean-Doubling:
  $$t_i = 2^{i-1}\mu,\ \forall i \geq 2$$
  - Mean-Stdev:
  $$t_i = \mu + (i-1)\sigma,\ \forall i \geq 2$$
  - Median-by-Median:
  $$t_i = Q(1 - \frac{1}{2^i}),\ \forall i \geq 2$$

Methodology

- Expected cost of sequence approximated via Monte-Carle process

- Expected cost of sequence approximated via Monte-Carle process
- Normalization by *omniscient scheduler*

$$\mathbb{E}^o = \int_0^\infty (\alpha t + \beta t + \gamma) f(t) dt = (\alpha + \beta) \cdot \mathbb{E}[X] + \gamma$$

- Expected cost of sequence approximated via Monte-Carle process
- Normalization by *omniscient scheduler*

$$\mathbb{E}^o = \int_0^\infty (\alpha t + \beta t + \gamma) f(t) dt = (\alpha + \beta) \cdot \mathbb{E}[X] + \gamma$$

- Evaluation over two reservation-based scenarios:
  - RESERVATIONONLY: "pay what you request": $\alpha = 1,\ \beta = \gamma = 0$
  - NEUROHPC:
    1. waiting time: $\beta = 1$, (α, γ) by curve fitting waiting time from platform data
    2. execution time: neuroscience application fitting

ReservationOnly: General heuristics performance

Distribution	Brute-Force	Mean-by-Mean	Mean-Stdev	Mean-Doub.	Med-by-Med	Equal-time	Equal-prob.
Gamma							
Uniform							
TruncatedNormal							
Beta							

Table: Normalized expected costs. Value in brackets = expected costs normalized by Brute-Force performance.

ReservationOnly: General heuristics performance

Distribution	Brute-Force	Mean-by-Mean	Mean-Stdev	Mean-Doub.	Med-by-Med	Equal-time	Equal-prob.
Gamma	2.02	2.45 (1.21)	2.26 (1.12)	2.22 (1.10)	2.66 (1.31)	2.14 (1.06)	2.09 (1.03)
Uniform							
TruncatedNormal							
Beta							

Table: Normalized expected costs. Value in brackets = expected costs normalized by Brute-Force performance.

ReservationOnly: General heuristics performance

Distribution	Brute-Force	Mean-by-Mean	Mean-Stdev	Mean-Doub.	Med-by-Med	Equal-time	Equal-prob.
Gamma	2.02	2.45 (1.21)	2.26 (1.12)	2.22 (1.10)	2.66 (1.31)	2.14 (1.06)	2.09 (1.03)
Uniform	1.33	2.21 (1.66)	1.86 (1.40)	1.69 (1.27)	2.22 (1.67)	1.33 (1.00)	1.33 (1.00)
TruncatedNormal							
Beta							

Table: Normalized expected costs. Value in brackets = expected costs normalized by Brute-Force performance.

ReservationOnly: General heuristics performance

Distribution	Brute-Force	Mean-by-Mean	Mean-Stdev	Mean-Doub.	Med-by-Med	Equal-time	Equal-prob.
Gamma	2.02	2.45 (1.21)	2.26 (1.12)	2.22 (1.10)	2.66 (1.31)	2.14 (1.06)	2.09 (1.03)
Uniform	1.33	2.21 (1.66)	1.86 (1.40)	1.69 (1.27)	2.22 (1.67)	1.33 (1.00)	1.33 (1.00)
TruncatedNormal	1.34	1.96 (1.46)	1.83 (1.36)	2.02 (1.50)	2.17 (1.61)	1.36 (1.01)	1.38 (1.03)
Beta	1.75	2.06 (1.18)	2.09 (1.19)	1.93 (1.10)	2.48 (1.42)	1.80 (1.03)	1.77 (1.01)

Table: Normalized expected costs. Value in brackets = expected costs normalized by Brute-Force performance.

ReservationOnly: RI or OD?

Reserved-Instance better than *On-demand*:

$$c_{RI} \cdot \tilde{\mathbb{E}}(S) \leq c_{OD} \cdot \mathbb{E}(X)$$
$$\frac{\tilde{\mathbb{E}}(S)}{\mathbb{E}(X)} \leq \frac{c_{OD}}{c_{RI}}$$

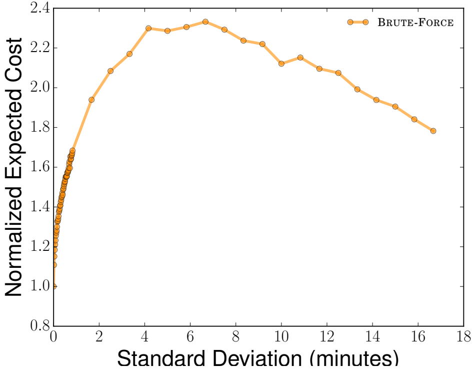


Figure: Truncated Normal distribution when σ varies from 0 to 2μ ($\mu \approx 8$min).

ReservationOnly: Discretization performance

Distribution	Equal-time							Equal-probability						
	$n = 10$	$n = 25$	$n = 50$	$n = 100$	$n = 250$	$n = 500$	$n = 1000$	$n = 10$	$n = 25$	$n = 50$	$n = 100$	$n = 250$	$n = 500$	1000
Exponential														
Lognormal														
Weibull														
Pareto														
Uniform														

Table: Normalized expected costs of the two discretization-based heuristics with different numbers of samples in the ReservationOnly scenario.

ReservationOnly: Discretization performance

Distribution	Equal-time							Equal-probability						
	$n = 10$	$n = 25$	$n = 50$	$n = 100$	$n = 250$	$n = 500$	$n = 1000$	$n = 10$	$n = 25$	$n = 50$	$n = 100$	$n = 250$	$n = 500$	1000
Exponential	2.64	2.32	2.43	2.49	2.28	2.39	2.33	3.66	2.88	2.35	2.41	2.35	2.32	2.43
Lognormal	2.02	1.92	1.97	1.93	1.90	1.93	1.89	2.99	2.32	2.13	1.99	1.87	1.93	1.99
Weibull														
Pareto														
Uniform														

Table: Normalized expected costs of the two discretization-based heuristics with different numbers of samples in the ReservationOnly scenario.

ReservationOnly: Discretization performance

Distribution	Equal-time							Equal-probability						
	$n = 10$	$n = 25$	$n = 50$	$n = 100$	$n = 250$	$n = 500$	$n = 1000$	$n = 10$	$n = 25$	$n = 50$	$n = 100$	$n = 250$	$n = 500$	1000
Exponential	2.64	2.32	2.43	2.49	2.28	2.39	2.33	3.66	2.88	2.35	2.41	2.35	2.32	2.43
Lognormal	2.02	1.92	1.97	1.93	1.90	1.93	1.89	2.99	2.32	2.13	1.99	1.87	1.93	1.99
Weibull	17.00	7.15	4.45	3.33	2.49	2.56	2.44	18.69	9.03	5.14	3.60	2.88	2.47	2.57
Pareto	31.54	13.02	6.84	3.79	2.12	1.75	1.74	35.49	11.73	9.99	5.97	2.89	2.59	1.78
Uniform														

Table: Normalized expected costs of the two discretization-based heuristics with different numbers of samples in the ReservationOnly scenario.

ReservationOnly: Discretization performance

Distribution	Equal-time							Equal-probability						
	$n=10$	$n=25$	$n=50$	$n=100$	$n=250$	$n=500$	$n=1000$	$n=10$	$n=25$	$n=50$	$n=100$	$n=250$	$n=500$	1000
Exponential	2.64	2.32	2.43	2.49	2.28	2.39	2.33	3.66	2.88	2.35	2.41	2.35	2.32	2.43
Lognormal	2.02	1.92	1.97	1.93	1.90	1.93	1.89	2.99	2.32	2.13	1.99	1.87	1.93	1.99
Weibull	17.00	7.15	4.45	3.33	2.49	2.56	2.44	18.69	9.03	5.14	3.60	2.88	2.47	2.57
Pareto	31.54	13.02	6.84	3.79	2.12	1.75	1.74	35.49	11.73	9.99	5.97	2.89	2.59	1.78
Uniform	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33

Table: Normalized expected costs of the two discretization-based heuristics with different numbers of samples in the ReservationOnly scenario.

NEUROHPC SCENARIO RESULTS

Instantiation from Neuroscience app (LogNormal distrib) + Intrepid waiting time cost.

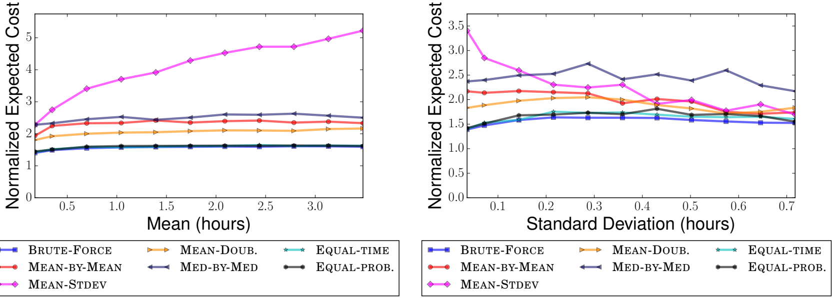


Figure: Impact of the mean or standard deviation in NEUROHPC scenario ($\mu = 7.1128, \sigma = 0.2039$) with $\alpha = 0.95,\ \beta = 1.0,\ \gamma = 1.05$.

Perspectives

- Contributions
  - Existence of optimal reservation sequence
  - Characterization up to duration of first reservation t_1^o
  - Upper-bound on t_1^o
  - Heuristics and comprehensive simulation results

- Future works
  - Requests with variable amount of resources (time + # processors)
  - Checkpoints at the end of some/all reservations
  - Trade-off
    - useful works under reservations
    - sacrifice of time to avoid losing all curent reservation work

Thanks